ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»
Факультет Санкт-Петербургская школа физико-математических и компьютерных наук

Масальский Даниил Вячеславович

РАСПОЗНАВАНИЕ РАБОЧИХ МЕСТ И ДИНАМИЧЕСКИЙ АНАЛИЗ ИХ ЗАНЯТОСТИ

Выпускная квалификационная работа
по направлению подготовки 01.04.02 Прикладная математика и информатика
образовательная программа «Машинное обучение и анализ данных»

Рецензент
бак., Student Research Assistant
_____
А.Н. Панфилов

Научный руководитель
к. ф-м. наук, проф.
_____
М.С. Мухин

Консультант
бак., рук. команды
_____
А.А. Сердюков

Санкт-Петербург 2023

Содержание

Аннотация

Данная работа имеет практический характер и посвящена распознаванию объектов с высоким уровнем абстракции (в данной работе конкретно распознаванию рабочих мест) для применения на камерах видеонаблюдения.

В этой работе описан полный процесс решения данной задачи: начиная со сбора и разметки данных, заканчивая интеграцией модели в систему для видеоаналитики.

Работа предполагает использование алгоритмов YOLOv5 и YOLOv8 для дообучения их на небольшом количестве подобранных данных. Данная работа показывает, что для решения такой задачи вполне может хватить небольшого количества данных.

Значимость этой работы заключена в ее результатах и в получении готового процесса для распознавания различных объектов разной степени абстракции.

Список ключевых слов

1. Распознавание объектов
2. Камеры видеонаблюдения
3. Компьютерное зрение
4. Рабочие места
5. Распознавание людей

Введение

Данная работа относится к классу работ посвященных задаче детекции объектов на фотографии. Однако ее практическая цель вносит специфику и выделяет ее на фоне классических, более “стерильных” научных работ. Две ключевые особенности усложняют применение обычных методов детекции в данном случае.

Одна из них это сама доменная область, а именно фото и видеоматериалы с камер наблюдения. Видеокамеры низкой и средней ценовой категории рассчитаны на использование в первую очередь человеком и хранение большого количества видеоматериалов. Это служит причиной того что качество видеозаписи крайне низкое, в темное время суток камера не захватывает цвета, сама видеозапись страдает от шума и алгоритмы шумоподавления встроенные в камеры зачастую размывают движущиеся объекты. Этот факт и нестандартные ракурсы камер существенно отличает данные от стерильных датасетов, на которых тренируют стандартные алгоритмы детекции. Это затрудняет их использование и требует некоторой доработки как данных так и самих алгоритмов.

Вторая сложность рассматриваемая данной работой – это детекция нестандартного высокоуровневого класса объектов, а именно – рабочих мест. Основная сложность детекции такого класса в том, что даже человеку сложно дать определение этому классу, так как в контексте каждого человека оно может иметь разные значения. Данная работа рассматривает по большей части применение алгоритмов детекции именно к классу рабочих мест, однако в будущем может быть обобщена на высокоуровневые абстракции целиком.

1. Обзор предметной области

Этот раздел будет посвящен анализу существующих на рынке или в общем доступе решений для задач детекции, особенно мы уделим внимание решениям позволяющим детектировать/сегментировать сложные и возможно композитные объекты, которые так важны для данной работы.

1.1. OWL-ViT

OWL-ViT[1] или Visual Transformer for Open-World Localization это работа посвященная детекции объектов из открытого словаря. То есть она сочетает в себе обработку текстового входа и обработку изображения двумя схожими по архитектуре моделями. Таким образом полученная авторами модель способна решать очень важную задачу open-dictionary object detection, что подразумевает поиск таких объектов, которые модель не видела на стадии обучения локализации объектов в пространстве.

Данная модель использует трансформерные архитектуры для получения векторного представления как текста так и изображения. Как базовая модель для энкодера изображения используется трансформер для изображений ViT слегка модифицированный для задачи детекции объектов. Для получения текстового представления используется схожий по архитектуре трансформер принимающий на вход текст для детекции.

Тренировка этих моделей происходит в 2 этапа. Сначала проводится общая предтренировка на большом количестве картинок из интернета и описаний к ним. Такая предтренировка помогает совместить выходы текстового энкодера и энкодера изображений. Для этого авторы используют contrastive loss[2], необходимый для

совмещения показаний выходных голов аттеншена двух различных моделей.

Второй (основной) этап тренировки уже использует синхронизированные выходные головы аттеншена для локализации текстовых запросов на фото. На этом этапе используются стандартные для задачи детекшена датасеты. Как текстовый вход используется название класса, на вход визуальному трансформеру подается картинка и учится модель на предсказание ограничивающих боксов для каждого объекта. В итоге получается что каждая выходная голова аттеншена должна сосредоточится на конкретном объекте и получить информацию важную как для его класса так и для его локализации.

На рисунке 1 приведена схема обучения для модели OWL-ViT на двух этапах:

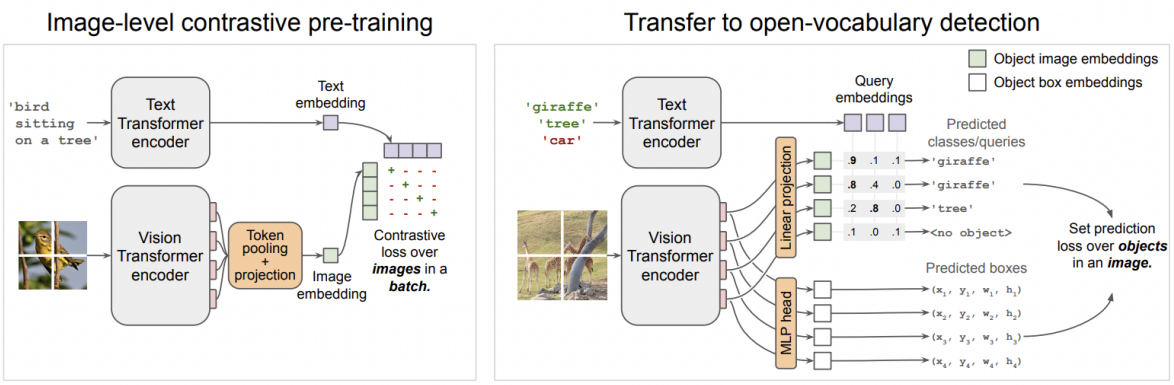

Рисунок 1 – два этапа обучения модели OWL-ViT

Данная модель используется в этой работе для проверки гипотезы о том, что модель заточенная под текстовые эмбеддинги будет хорошо предсказывать и локализовать сложные абстрактные классы.

1.2. YOLOv5/YOLOv8

Работы YOLOv5[3] и YOLOv8[4] придуманы и разработаны одними и теми же авторами и являются идейными продолжателями модели YOLO[5], которая в свое время показала, что задача детекции объектов может быть решена как задача регрессии и при этом со скоростью позволяющей обработку кадров с видеозаписей в реальном времени.

Данные модели выходили с промежутком в несколько лет, однако как показывает практика, 8 версия не так уж далеко ушла от 5. Обе модели предсказывают объекты схожим с оригинальным YOLO способом, однако имплементируют некоторые доработки улучшающие, как сходимость во время тренировки, так и итоговое качество детекции.

Из конкретных доработок, обе модели используют якорные позиции для более точной локализации объектов. Эти якорные объекты получаются алгоритмами кластеризации на тренировочных данных, модели затем предсказывают не конкретные позиции объектов, а лишь отклонения от якорных объектов.

Данные модели используются как основные модели для данной работы, так как их скорость позволяет им обрабатывать кадры с видеокамер в реальном времени даже на CPU, что является основным ограничением данной работы.

1.3. Real-Time Abnormal Object Detection for Video Surveillance in Smart Cities

В качестве работы проводимой в схожем направлении рассмотрим работу под названием Real-Time Abnormal Object Detection for Video Surveillance in Smart Cities[6], она интересна именно с доменной

стороны, так как авторы решают задачу детекции объектов будучи ограниченными в качестве изображений на входе.

Данная работа сконцентрирована на детекции оружия и других мелких опасных предметов в реальном времени с помощью алгоритмов глубокого обучения. Авторы подмечают что классические алгоритмы компьютерного зрения сильно подвержены воздействию шума и поэтому отвергаются в данной работе.

Так как авторов интересует детекция мелких опасных объектов (таких как ножи или пистолеты) то зашумление изображения сильно влияет на качество. Однако в данной работе они берут за основу для детекции такие модели как VGG[7] и не рассматривают ни одного способа сглаживания или исправления шумных изображений.

На рисунке 2 приведен процесс принятия решений в работе Real-Time Abnormal Object Detection for Video Surveillance in Smart Cities. Стоит заметить, что явного этапа для борьбы с шумом на картинке с камеры видеонаблюдения авторы не выделяют.

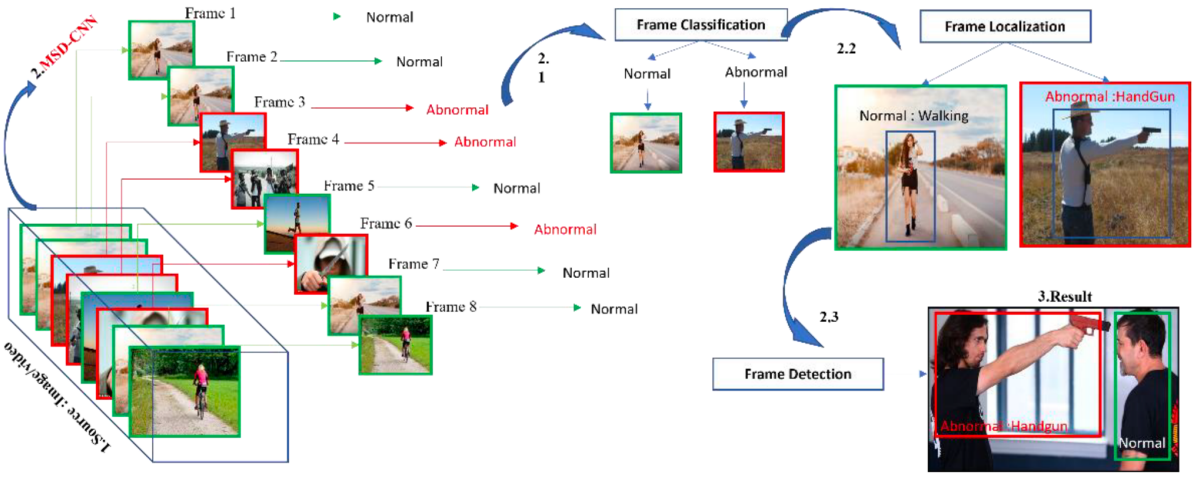

Рисунок 2 – Процесс применения алгоритмов в работе Real-Time Abnormal Object Detection for Video Surveillance in Smart Cities

Для моей работы важность Real-Time Abnormal Object Detection for Video Surveillance in Smart Cities заключается в актуальности проблемы детекции изображений с камер наблюдения и в отсутствии адекватных способов борьбы с низким качеством изображений даже в публикациях 2022 года.

1.4. Выводы по главе

Данная глава подчеркивает актуальность этой работы, так как существует много современных алгоритмов способных совершать распознавание объектов с высокой абстракцией в реальном времени. Однако доменная область распознавания объектов с камер видеонаблюдения существенно отстает от этих результатов и даже в актуальных публикациях используются старые, вышедшие из использования в других областях, методы.

2. Сбор данных

Для данной работы в первую очередь был необходим датасет, позволяющий дообучать на нем классические модели для детекции объектов. Так как решаемая задача включает в себя детекцию с высоким уровнем абстракции, то также было необходимо составить описание задачи для разметчиков.

2.1. Разметка людей

Разметка людей производилась с помощью краудсорсинг платформы Yandex.Toloka. Данные для разметки были собраны с видеокамер расположенных в Университете ИТМО.

Для упрощения задачи для разметчиков, а также из соображений экономии была предпринята попытка сделать некоторую предразметку с помощью мощных алгоритмов детекции, не способных работать в реальном времени. В качестве такого алгоритма был выбран Detectron2[8]. Это действительно удешевило сбор данных, но и вызвало некоторые проблемы с разметкой, так как разметчики не сразу понимали суть задачи, часто пропускали инструкцию или же считали, что вся работа за них уже сделана и никак не меняли разметку.

В итоге после некоторых итераций (в ходе которых размер датасета сократился до 9500 фотографий) разметка по людям была получена.

Итоговый размер датасета составил 9500 фотографий. Датасет получился достаточно качественный и будет использоваться для тренировки и валидации алгоритмов в этой работе.

Итоговый интерфейс задачи для разметчиков приведен ниже на рисунке 3:

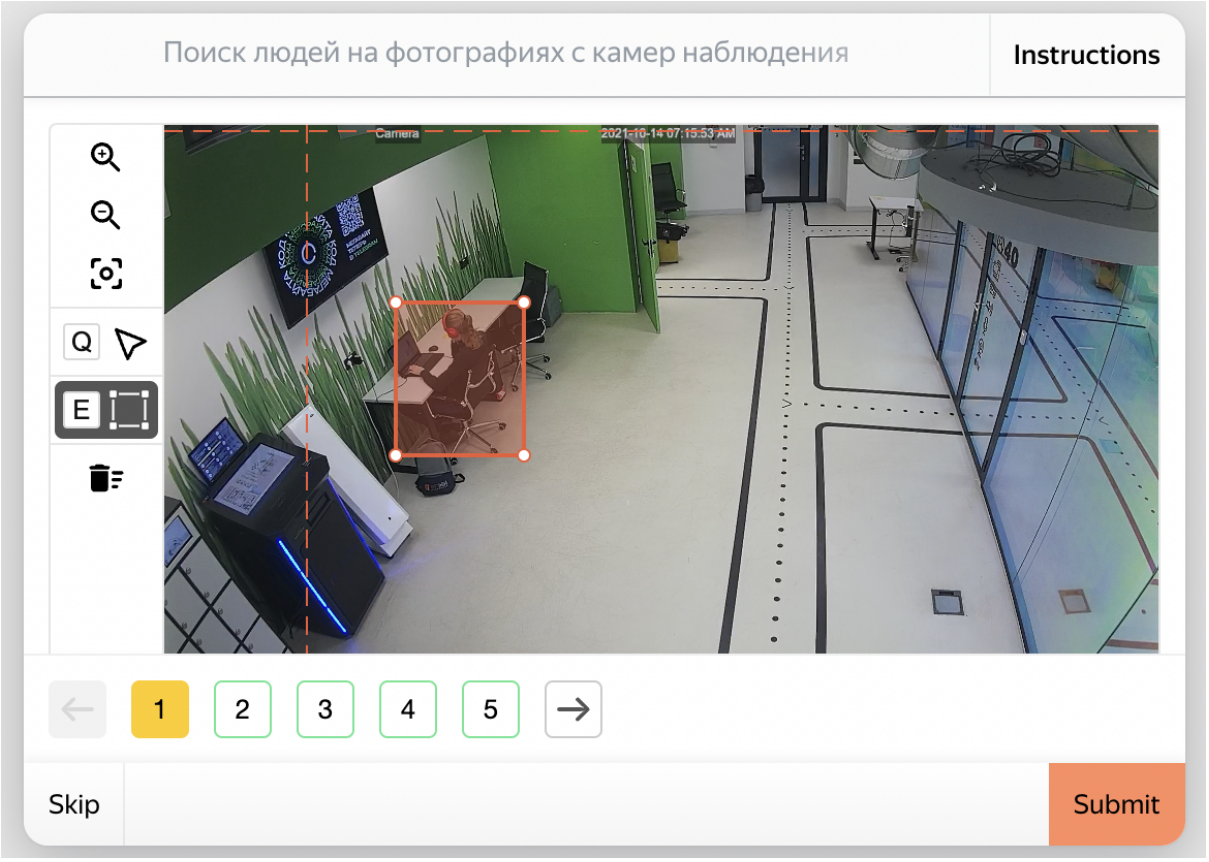

Рисунок 3 – Интерфейс разметчика для задачи на сервисе Яндекс.Толока

Так как главное условие при составлении разметки это подробное описание задачи и множество примеров так же было составлено подробное руководство для разметчиков, где были формализованы сразу все необходимые правила для разметки. Пример из руководства приведен ниже на рисунке 4:

Рисунок 4 – Пример из инструкции для разметчиков

2.2. Разметка рабочих мест

Для данной задачи в первую очередь было сформулировано определение рабочего места. Таким местом решено было назвать “место где человек может расположиться для работы с персональным ноутбуком или с установленным на рабочем месте оборудованием”.

Так как при сборе предыдущего датасета были сделаны выводы о среднем качестве разметки в краудсорсинге и учитывая прогрессию в сложности по сравнению с предыдущей задачей, было решено провести разметку рабочих мест вручную с помощью сервиса Dataloop.

В силу этого размер датасета получился относительно маленький, всего 600 фотографий, однако на каждой фотографии было в среднем по 5-6 рабочих мест, чего при должном количестве аугментаций вполне достаточно чтобы хоть немного научить модель.

Рабочие места были размечены как занятые и свободные в зависимости от наличия визуальных признаков их занятости (верхняя одежда на спинке стула или же стоящий на столе ноутбук относят рабочее место к занятым).

2.3. Выводы по главе

Данная глава описывает инструменты и подходы используемые для сбора и разметки данных. Помимо этого в ней описаны какие типы данных были собраны и как данная работа формализует понятие “рабочее место”.

3. Конфигурация экспериментов

3.1. Выбранные модели

В качестве моделей для проведения сравнительного анализа были выбраны 3 модели, 2 схожие между собой YOLOv5 и YOLOv8 и 1 модель OWL-ViT как представитель класса визуальных трансформеров.

Обе модели из семейства YOLO представляют особый интерес для данной работы, так как позволяют обрабатывать стримы с камер видеонаблюдения в реальном времени. Это одно из необходимых ограничений данной работы, поэтому упор ставится именно на обучение этих двух моделей.

Визуальный трансформер не позволит обрабатывать кадры в реальном времени, однако сам его подход к решению задачи стоит того чтобы рассмотреть его в данной работе. Особенность OWL-ViT в том что она позволяет принимать описание искомого класса как текстовый вход. В данной работе исследуются ее возможности в поиске сложных объектов с высоким уровнем абстракции.

3.2. Что проверяют эксперименты

Формализуем задачи, которые ставит перед собой данная работа. Назовем рабочим местом такое место, в котором человек мог бы расположиться со своим ноутбуком для проведения рабочей встречи/выполнения рабочих задач.

Вторая задача данной работы заключается в том, что большинство записей с видеокамер имеют совершенно разные типы шума, которые включают в себя как и простые выбросы цвета, так и деформацию кадра более сложного вида – размытие кадров вследствие работы алгоритмов

шумоподавления. Так как данная работа подразумевает использование ее на стримах с камер различного вида, то нет возможности подготовиться к какому-то типу шума. Так что вторая часть работы заключается в попытке применить шумоподавление для повышения качества работы.

Перейдем к подзадачам: детекция людей необходима для проверки того, что модели адаптировались к данным, плюс это очень полезная подзадача используемая в дальнейшем для решения проблем данной работы.

Вторая подзадача – классификация найденного рабочего места на занято/свободно. Данная задача появляется из изначального дизайна системы, который предполагает использование независимых моделей для детекции и классификации объектов. Такое предположение делается из-за того что вероятно умное распознавание рабочих мест будет сильно неэффективно и придется использовать статическую локализацию, а классификация может выполняться и при статической локализации в отличие от совмещенной модели для детекции и классификации.

Третья подзадача заключается в динамическом маппинге рабочих мест между соседними кадрами. Для этого предполагается написать простой алгоритм.

Последняя подзадача состоит в том чтобы имея временной ряд для каждого рабочего места определить единовременную занятость рабочего места одним человеком. Для этих целей предполагается использовать одномерные морфологические методы.

Ниже на рисунке 5 приведена схема, иллюстрирующая принцип работы итоговой системы, этот принцип сохраняется и для алгоритма обучения в том числе.

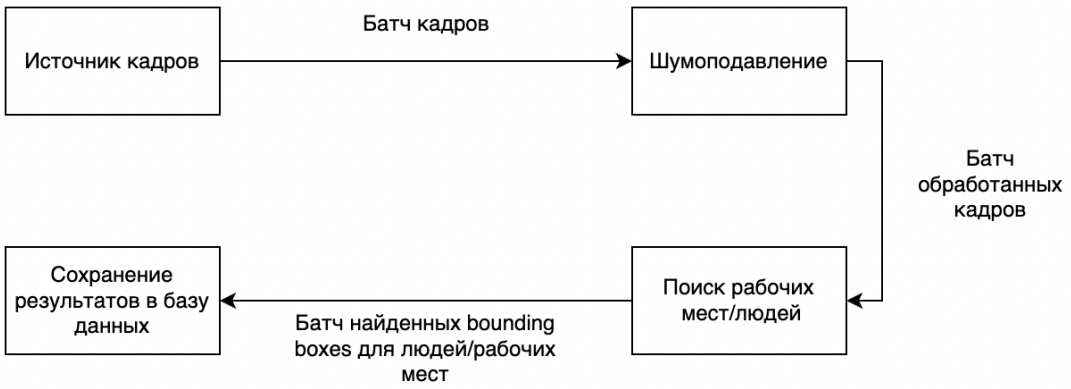

Рисунок 5 – Принцип работы системы распознавания людей/рабочих мест

3.3. Конфигурация экспериментов для поиска объектов

Для детекции людей эксперименты проводились следующим образом: 6 моделей обучались на 8.5 тысячах фотографий и валидировалась на 1 тысяче фотографий. Каждая модель была обучена в двух конфигурациях: с подавлением шума и без. Для подавления шума использовалась свертка с Гауссовским фильтром[9].

Иллюстрация работы Гауссовского фильтра приведена на рисунке 6, однако стоит отметить то, что гауссовский фильтр в данной работе имеет меньшую сигму и за счет этого теряет меньше информации и не приводит изображение к столь размытому состоянию как в примере.

Рисунок 6 – пример работы гауссовского фильтра

Помимо 6 моделей которые были дообученны на специальных данных с камер наблюдения для детекции людей был использована также базовая версия визуального трансформера с текстовым входом OWL-ViT. Для улучшения ее результатов проводился подбор промптов и были выбраны несколько оптимальных (“Person”, “Human”, “Human being”).

Так же для сравнения прогресса моделей будут приведены сравнения базовых то есть не дообученных версий YOLO.

Для детекции рабочих мест проводились те же самые эксперименты, однако размер данных для обучения был значительно меньше (500 изображений). Так же использовалась свертка с Гауссовским фильтром для подавления шума в изображениях.

Как и в случае с детекцией людей на изображении для визуального трансформера был проведен подбор оптимальных для данной задачи промптов и в итоге были выбраны несколько лучших (“Laptop workstation”, “Workstation”, “Workspace”). В данной конфигурации эксперименты качество модели также позволяет улучшить сильное

занижение трешхолда по уверенности модели и использование затем на полученных локализациях операции non-maximum suppression[10].

Так как базовых моделей для детекции рабочих мест YOLO не имеет то и сравнений с такими моделями не будет. Однако для детекции рабочих мест можно подобрать свой бейзлайн. Таким бейзлайном будет фиксированная локализация рабочих мест для каждой камеры. Этот подход ошибается в случае частых перестановок рабочих мест в коворкингах или открытых пространствах и такие феномены хорошо представлены в имеющемся датасете.

3.4. Конфигурация экспериментов для классификации объектов

Эксперименты с классификацией подразумевают у нас наличие фотографии и отмеченных на ней рабочих мест. Имея такой вход система должна решить для каждого найденного рабочего места его класс, а точнее занято оно или же свободно.

Данная работа предлагает 2 решения для данной проблемы. Первое – для каждой фотографии проводить еще и поиск людей на ней и затем проверять пересечение локализованных людей с рабочими местами и определять занятость с помощью установленного трешхолда по пересечению. Пересечение предлагается определять с помощью метрики Intersection Over Union[11].

Процесс описанный в первом решении проиллюстрирован на рисунке 7, все комментарии к процессу приведены на нем.

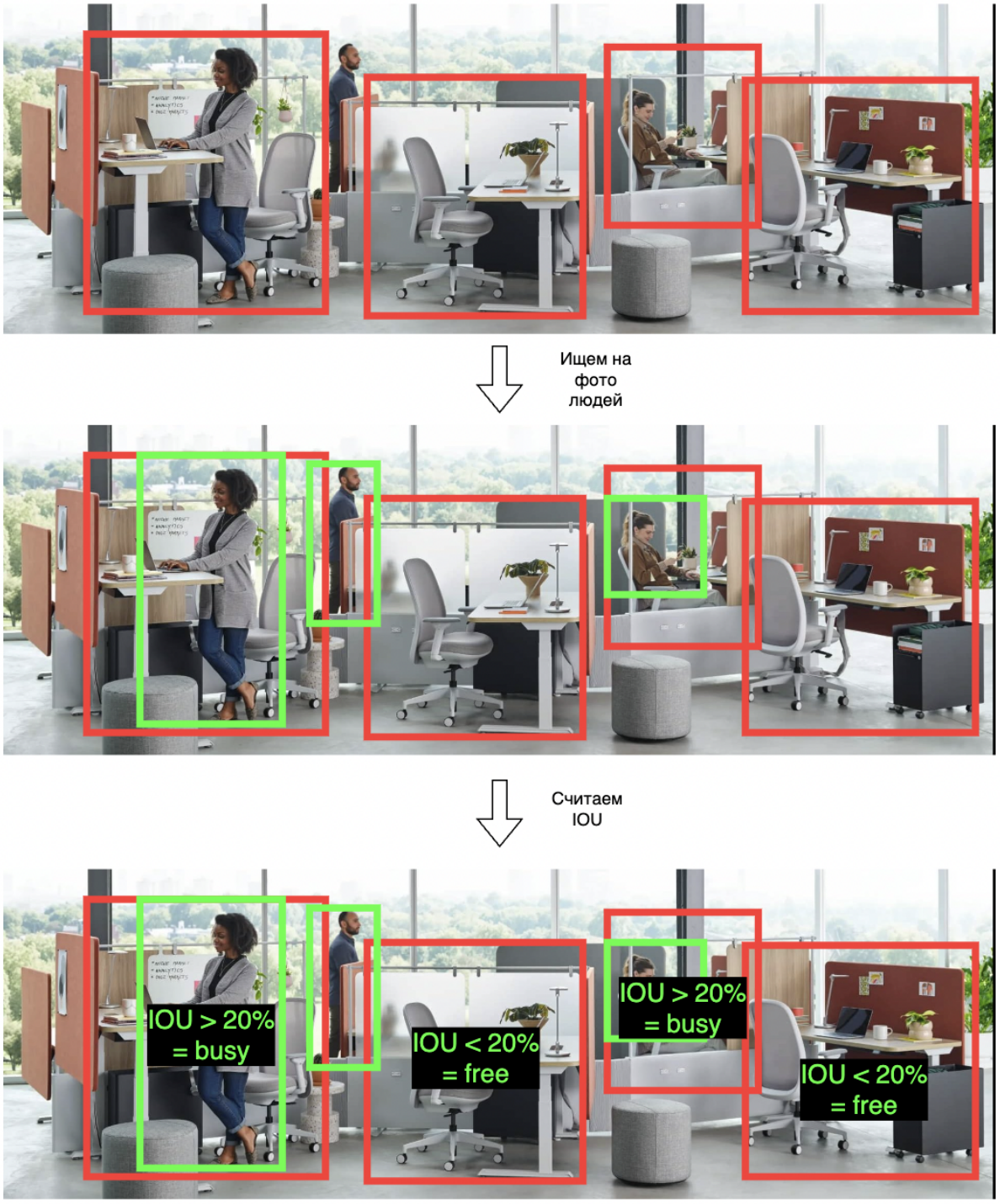

Рисунок 7 – Процесс классификации рабочих мест на занятые и свободные: первый способ

Второе решение – вырезать все найденные рабочие места с картинки и обработать их с помощью отдельной сверточной нейросети. Поэтому для такого решения нам нужно иметь обученный

классификатор. Он дообучается на известных нам данных. Затем используется на вырезанных фрагментах для проверки.

Процесс описанный во втором решении проиллюстрирован на рисунке 8, комментарии к процессу так же приведены на рисунке.

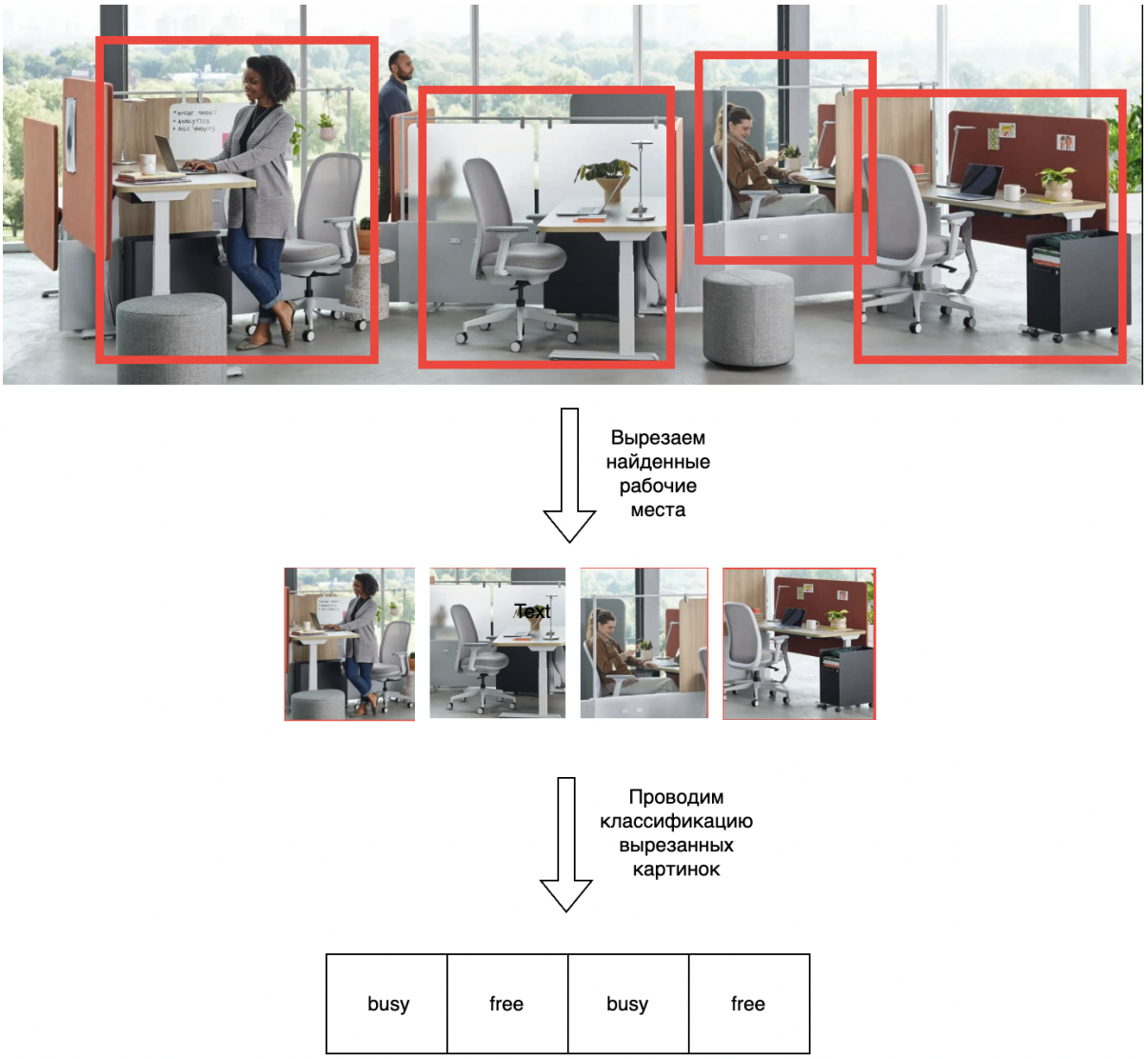

Рисунок 8 – Процесс классификации рабочих мест на занятые и свободные: второй способ

3.5. Выводы по главе

Данная глава описывает постановку и конфигурацию экспериментов для различных моделей и задач. Все эти задачи объединены общей задачей распознавания объектов. Однако еще в ней описаны подходы к решению задачи классификации большого количества объектов в реальном времени.

4. Результаты экспериментов

4.1. Поиск объектов

Все эксперименты проводились в соответствии с конфигурацией описанной в пункте 3 выше. Результаты экспериментов с распознаванием людей на изображениях с видеокамеры представлены в таблице 1.

Модель	AP@50 обычные фото	AP@75 обычные фото	AP@50 с подавлением шума	AP@75 с подавлением шума
YoloV5n finetuned	0.632 +- 0.008	0.311 +- 0.009	0.656 +- 0.01	0.342 +- 0.009
YoloV5s finetuned	0.663 +- 0.009	0.363 +- 0.01	0.686 +- 0.009	0.367 +- 0.009
YoloV5m finetuned	**0.699 +- 0.01**	**0.442 +- 0.01**	**0.706 +- 0.01**	**0.437 +- 0.011**
YoloV8n finetuned	0.639 +- 0.009	0.403 +- 0.01	0.658 +- 0.009	0.386 +- 0.009
YoloV8s finetuned	0.636 +- 0.01	0.392 +- 0.01	0.65 +- 0.01	0.427 +- 0.01
YoloV8m finetuned	0.645 +- 0.009	0.417 +- 0.01	0.655 +- 0.009	0.435 +- 0.011
OWL-ViT	0.211 +- 0.007	0.071 +- 0.004	0.289 +- 0.008	0.094 +- 0.005
YoloV5s	0.451 +- 0.012	0.275 +- 0.01	0.476 +- 0.009	0.291 +- 0.01
YoloV8s	0.468 +- 0.01	0.336 +- 0.009	0.489 +- 0.009	0.354 +- 0.009

Таблица 1 – Результаты экспериментов с распознаванием людей

Глядя на эти результаты можно сделать несколько выводов:

1. Разницы между поколениями YOLOv5 и YOLOv8 практически не заметно, среди них нет лучшей во всем модели;
2. Лучшая модель – это YOLOv5m, она превосходит своих конкурентов во всех конфигурациях;
3. Трансформер OWL-ViT сильно проигрывает специализированным дообученным и даже базовым моделям в обеих конфигурациях;
4. Подавление шума улучшает метрики детекции почти для всех моделей, однако для некоторых не статистически значимо.

Результаты экспериментов с детекцией рабочих мест приведены ниже в таблице 2. Все эксперименты проводились в соответствии с описанием из пункта 3.3.

Модель	AP@50 обычные фото	AP@75 обычные фото	AP@50 с подавлением шума	AP@75 с подавлением шума
YoloV5n	0.13 +- 0.012	0.015 +- 0.004	0.152 +- 0.012	0.024 +- 0.007
YoloV5s	0.262 +- 0.012	0.063 +- 0.01	0.224 +- 0.012	0.045 +- 0.01
YoloV5m	**0.314+- 0.013**	**0.133 +- 0.016**	**0.299 +- 0.012**	**0.109+- 0.015**
YoloV8n	0.243 +- 0.016	0.07 +- 0.014	0.209+- 0.012	0.052 +- 0.011
YoloV8s	0.278 +- 0.016	0.099 +- 0.014	**0.291 +- 0.014**	**0.114 +- 0.019**
YoloV8m	0.305 +- 0.015	0.115 +- 0.013	**0.295 +- 0.013**	**0.108 +- 0.011**
OWL-ViT	0.069 +- 0.012	0.002 +- 0.001	0.054 +- 0.012	0.005 +- 0.004
Фиксированные места для определенных камер	0.156 +- 0.014	0.05 +- 0.012	0.156 +- 0.014	0.05 +- 0.012

Таблица 2 – Результаты экспериментов с распознаванием рабочих мест

Глядя на эти результаты можно сделать несколько выводов:

1. Разница между поколениями YOLOv5 и YOLOv8 практически не заметно, среди нет лучшей во всем модели;
2. Лучшая модель – это YOLOv5m, она статистически значимо превосходит своих конкурентов почти во всех конфигурациях;
3. Трансформер OWL-ViT сильно проигрывает специализированным дообученным моделям, его уникальные обобщающие способности на основе текстового входа не помогли ему даже близко подобраться к метрикам других моделей;
4. Подавление шума в данной ситуации может делать детекцию даже несколько хуже, причина этого требует дальнейшего изучения;
5. Бейзлайн решение с фиксированными рабочими местами для каждой камеры показало себя хуже чем специально дообученные модели.

Помимо экспериментов стоит также привести пример работы модели. На рисунке 9 проиллюстрированы рабочие места найденные на картинке, схожие с которой картинки встречались в тренировочных данных.

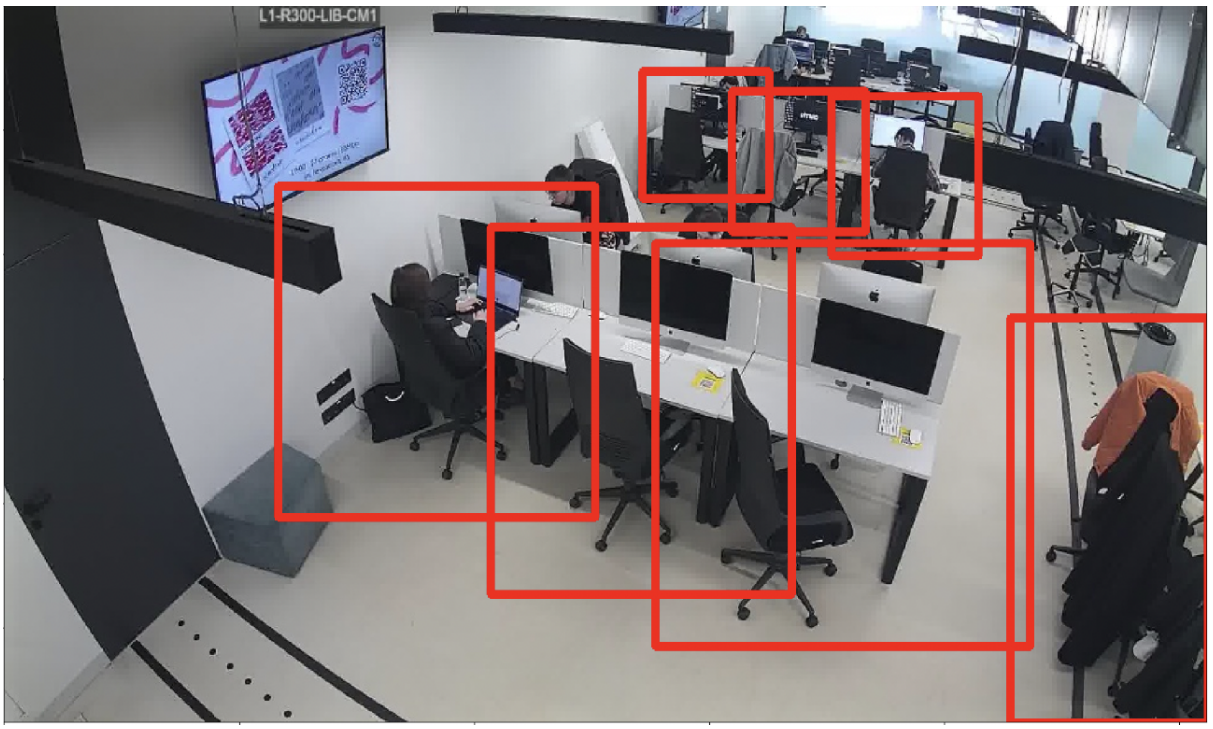

Рисунок 9 – Найденные моделью рабочие места на фото из тренировочных данных

Рисунок 10 иллюстрирует найденные моделью рабочие места на случайном фото из интернета. Можно сделать вывод, что некоторое обобщение знаний о рабочих местах модель все же сделала.

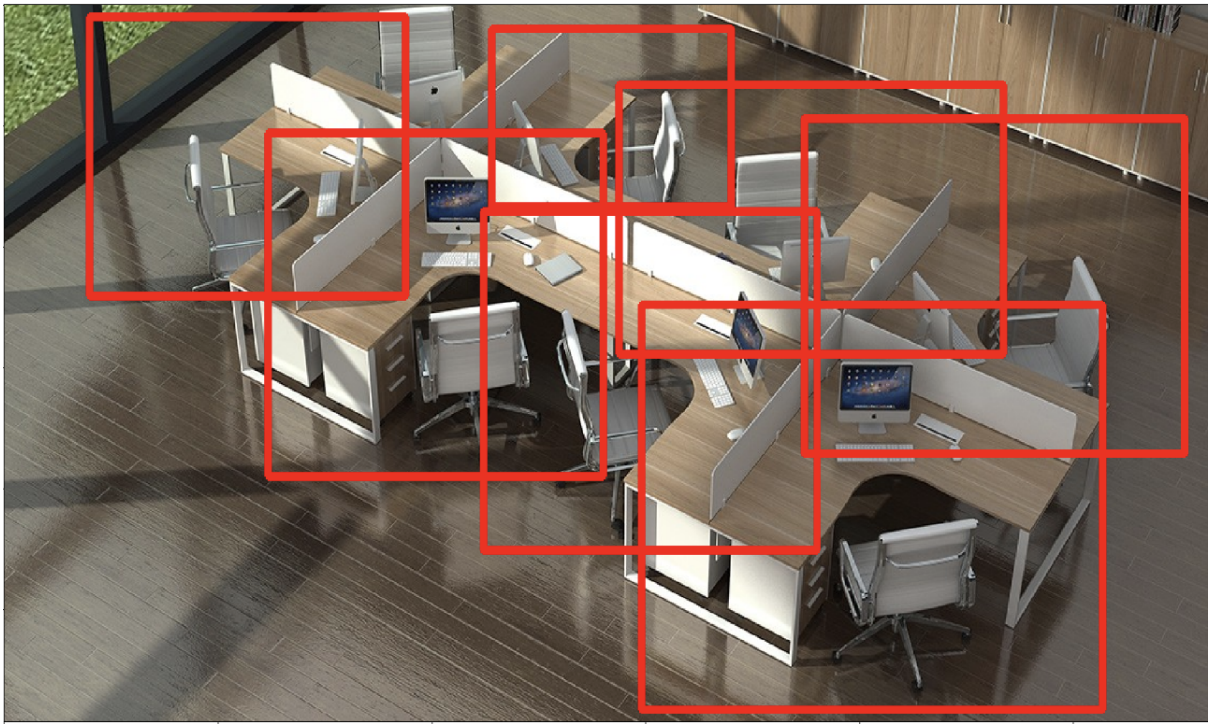

Рисунок 10 – Найденные моделью рабочие места на случайном фото из интернета

4.2. Классификация объектов

Все эксперименты проводились в соответствии с конфигурацией описанной в пункте 3.4 выше. Результаты экспериментов с классификацией рабочих мест на “занято” или “свободно” представлены ниже в таблице 3. Из-за несбалансированности датасета (примерно 70% объектов принадлежит классу “свободно”) большое значение имеют метрики Precision, Recall и F1-score.

Алгоритм	Accuracy	Presicion	Recall	F1-score
Пересечение с человеком	0.79	0.86	0.58	0.69
ResNet18	0.88	0.91	0.67	0.77

Таблица 3 – Результаты экспериментов с классификацией рабочих мест

Глядя на эти результаты можно сделать несколько выводов:

1. Специально обученная модель работает лучше наивного алгоритма;
2. Даже бейзлайн в виде пересечений дает достаточный для дальнейшей работы результат;
3. Бейзлайн решение проще в интеграции и в имплементации. Так же оно выигрывает и в скорости инференса. Проигрывает но не очень сильно по качеству специальной модели;

4.3. Выводы по главе

Данная глава описывает результаты экспериментов сконфигурированных в главе 3. Также в главе приведены основные выводы по результатам каждого эксперимента.

5. Аналитика на найденных местах

5.1. Межкадровый маппинг рабочих мест

Так как система детекции рабочих мест и классификации их занятости работает покадрово, то необходим алгоритм позволяющий соединить рабочие места между каждрами так чтобы на выходе получить последовательность классов для каждого рабочего места. Это необходимо для проведения дальнейшей аналитики.

Данная работа предлагает алгоритм для решения такой задачи. Алгоритм получает на вход последовательность кадров и найденных рабочих мест на каждом кадре. Затем алгоритм на каждом кадре находит центроиду каждого рабочего места. Теперь все что остается это для каждой центроиды одного кадра выбрать соответствующее ей рабочее место на следующем кадре. Для этого мы для каждой центроиды рабочего места выберем ближайшую по Евклидовому расстоянию центроиду на следующем кадре. Каждую цепочку таких центроид мы собираем в связке с временной отметкой этого места. Те центроиды для которых мы не нашли соответствие (евклидово расстояние выше заданного трешхолда) отправляются в пул проверки на следующем этапе (это нужно так как алгоритм детекции может пропустить рабочее место однако оно затем вновь найдется через несколько кадров).

Иллюстрация работы данного алгоритма приведена на рисунке 11.

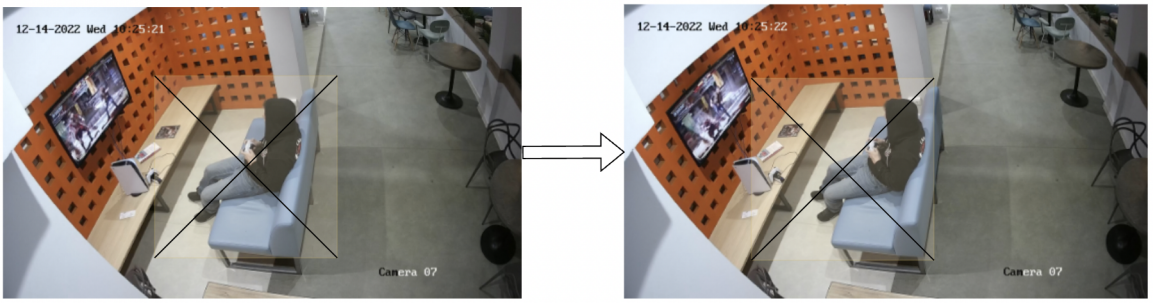

Рисунок 11 – Иллюстрация алгоритма маппинга рабочих мест между кадрами

Конечно такой алгоритм алгоритм имеет определенные недостатки и не стоит о них забывать пользуясь им, однако разработка более продвинутого алгоритма выходит за рамки данной работы.

5.2. Аналитика занятости рабочих мест

В пункте 5.1 мы смогли получить последовательность классов “занято”/”свободно” для каждого рабочего места. Текущий пункт будет посвящен дальнейшей обработке полученной информации.

Так у нас встает вопрос, как долго один человек занимал конкретное рабочее место. На этот вопрос мы бы могли ответить по последовательности из предыдущего пункта. Логика в следующем: человек занимает рабочее место затем место освобождается и занимается другим человеком снова. Пробел между ними позволяет нам понять что произошла смена человека на рабочем месте. Что же требуется сделать для того чтобы выделить такие пробелы и исправить ложные кратковременные пробелы. Мы можем применить популярные в компьютерном зрении морфологические методы чтобы сгладить такие ситуации.

В итоге алгоритм имеет следующий вид: получаем вход в виде последовательности занятости и незанятости рабочего места, проводим дилатацию и затем эрозию, получаем очищенную цепочку только с валидными последовательностями занятости и незанятости рабочего места.

5.3. Выводы по главе

Данная глава описывает дальнейшую аналитику производимую на полученных классифицированных на “занято”/”свободно” списков рабочих мест. А именно как проводить межкадровый маппинг рабочих мест из этих списков и как затем проводить аналитику занятости поверх полученных после маппинга рядов.

Заключение

Данная работа посвящена проблеме поиска сложных объектов с высоким уровнем абстракции на видеозаписях с камер видеонаблюдения. В ходе работы были рассмотрены несколько видов решений двух основных задач составляющих тему данной работы. Первая задача это поиск рабочего места и для нее были рассмотрены 3 различные модели с различными конфигурациями обучения и инференса. Эти модели: YOLOv5, YOLOv8, OWL-ViT. Для второй задачи, а именно классификации рабочих мест, было предложено 3 алгоритма, причем один из них включал в себя обучение модели классификатора на основе архитектуры ResNet18[12]. Все авторские модификации внесенные в модели на этапе препроцессинга и постпроцессинга описаны в соответствующих главах данной работы. Интегрированная система по результатам данной работы позволяет динамически в реальном времени определять рабочие места в коворкингах и открытых пространствах, классифицировать их на занятые и незанятые и автоматически выгружать отчеты с аналитикой для владельцев коворкингов или открытых пространств.

Библиографический список

1. Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby. (2022). Simple Open-Vocabulary Object Detection with Vision Transformers.

2. Chernyavskiy, A., Ilvovsky, D., Kalinin, P., and Nakov, P. 2022. Batch-Softmax Contrastive Loss for Pairwise Sentence Scoring Tasks. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 116–126). Association for Computational Linguistics.

3. Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, Kalen Michael, TaoXie, Jiacong Fang, imyhxy, Lorna, 曾逸夫(Zeng Yifu), Colin Wong, Abhiram V, Diego Montes, Zhiqiang Wang, Cristi Fati, Jebastin Nadar, Laughing, UnglvKitDe, Victor Sonck, tkianai, yxNONG, Piotr Skalski, Adam Hogan, Dhruv Nair, Max Strobel, and Mrinal Jain. (2022). ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation.

4. Jocher, G., Chaurasia, A., and Qiu, J.. (2023). YOLO by Ultralytics.

5. Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. (2016). You Only Look Once: Unified, Real-Time Object Detection.

6. Ingle, P., and Kim, Y.G. 2022. Real-Time Abnormal Object Detection for Video Surveillance in Smart Cities. Sensors, 22(10).

7. Simonyan, K., and Zisserman, A. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR, abs/1409.1556.

8. Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., and Girshick, R.. (2019). Detectron2. .

9. Spurek, P., Chaikouskaya, A., Tabor, J., and Zając, E. 2014. A local Gaussian filter and adaptive morphology as tools for completing partially discontinuous curves. Computer Information Systems and Industrial Management, p.559–570.

10. Neubeck, A., and Van Gool, L. 2006. Efficient Non-Maximum Suppression. In Proceedings of the 18th International Conference on Pattern Recognition - Volume 03 (pp. 850–855). IEEE Computer Society.

11. Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. (2019). Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression.

12. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. (2015). Deep Residual Learning for Image Recognition.